

REMARKS

In the Office Action mailed September 29, 2003, the Examiner noted that claims 1-14 were pending and rejected claims 1-14. Claims 1-14 remain pending for reconsideration which is requested. No new matter has been added. The Examiner's rejections are traversed below.

On page 2 of the Office Action the Examiner rejected all claims under 35 U.S.C. § 102 as anticipated by Wimble.

Wimble is directed to a goal oriented debugging system. This system operates by providing the user with high level tools and that allow a debugging problem to be broken down into simpler problems to be solved. As stated by Wimble:

Goal directed debugging is the technique of presenting the user with higher levels of tools--tools which present useful abstractions in debugging, and implementing those tools by breaking down a complex problem into simpler subproblems repeatedly until a simple set of problems results which can be solved using a set of primitive debugging services.

(Wimble, col. 7, lines 14-19)

The Examiner has particularly cited Wimble col. 7, line 9 - col. 8, line 33 as allegedly teaching the features of claims 1 and 14. This portion of the discussion of Wimble is directed at a discussion of Wimble figures 3-6. This text discusses the goal directed debugging as forming sets goals and sub goals, performing operations associated with these goals and sub goals and reporting to the user. The text also particularly discusses an example of operations that accomplish this where the example discusses running a program, determining a hypothesis about failure when the program crashes, instrumenting the program to allow the hypothesis to be tested and then determining the correctness of the hypothesis from the test results of the instrumentation.

In particular, Fig. 3 provides a flowchart showing the overall debugging scheme that comprises individual steps such as shown by flowcharts in Fig. 4-6. According to Fig. 3, the system receives a user request (86), forms a set of goals and sub-goals relying on contextual knowledge 88, performs these goals and sub-goals (90), and reports findings to the user 92. Fig. 4 provides a flowchart demonstrating a debugging problem solving approach in which hypothesis are developed and tested by the debugging system. Fig. 5 provides a continuation of the Fig. 4 flowchart, and demonstrates a debugging problem solving approach in which hypothesis are developed and tested by the debugging system. Finally, Fig. 6 provides a flowchart, which is a continuation of Fig. 5 and shows the development of backtracking information, if possible. The text cited by the Examiner contain lines listing up a set of sub-

goals, 1 - 11, to explain further about the operation shown in Figs. 4-6 and indicated respectively by numerals 50 - 84.

More importantly, Wimble provides a discussion that is at a high or abstract level and at one that is particularly not at the level of the present invention making Wimble inadequate at teaching the present invention. The present invention is directed at a test support system intimately involved at a much deeper level of detail, an object programming level of detail, than Wimble. Wimble is not involved with such a level, as stated by Wimble:

The discussion below is of goal directed debugging in terms of C++ computer language interfaces. Goal directed debugging, however, is in no way tied to the C++ computer language or, indeed, even to object oriented programming.

(Wimble, col. 7, lines 9 -12)

In fact, Wimble is at such a level of teaching that Wimble, in effect, teaches away from the present invention. As stated by Wimble:

The user remains thinking about the problem in his own knowledge domain and doesn't have to become highly knowledgeable in all these other areas. Further, the user doesn't even have to perform the tedious work wherein it may be difficult to avoid human error.

(Wimble, col. 7, lines 36-40)

This attention to a high or abstract level of teaching is evident throughout the Wimble discussion (see col. 7, lines 15, 26 and 51 and col. 8, line 29)

In contrast, the present invention, as noted above is directed to the more detailed object oriented programming level of testing where object level programming details such as class, subclass, class generation, class inheritance, definitions, etc. are involved in what is being claimed.

In particular, the present invention is directed to a test support system in which "a test support class generation unit obtaining screen definition information about a test target screen program, and generating a test support class which is a subclass inheriting a class of the test target screen program according to the screen definition information, and a class for testing the test target screen program" (see claims 1 and 14). Wimble does not address much less teach or suggest such a unit.

It is submitted that the present claimed invention patentably distinguishes over Wimble and withdrawal of the rejection is requested.

The dependent claims depend from the above-discussed independent claims and are patentable over the prior art for the reasons discussed above. The dependent claims also recite additional features not taught or suggested by the prior art. For example, claim 2 calls for "generating a test specification for the test target screen program according to the screen

definition information" and Wimble says and suggests nothing about a test specification for a screen definition. The remaining dependent claims also emphasize features not taught or suggested by Wimble. It is submitted that the dependent claims are independently patentable over the prior art.

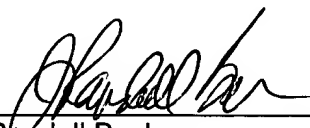
It is further submitted that the claims are not taught, disclosed or suggested by the prior art. The claims are therefore in a condition suitable for allowance. An early Notice of Allowance is requested.

If any further fees, other than and except for the issue fee, are necessary with respect to this paper, the U.S.P.T.O. is requested to obtain the same from deposit account number 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 12/29/13

By: 
J. Randall Beckers
Registration No. 30,358

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501